

# FLUTTER VS. REACT NATIVE – WHICH ONE TO CHOOSE IN 2022

In today's tech world, it's easy to embrace the latest trends, especially when new frameworks emerge with new pre-coded modules and guaranteed super-fast speed. Today, mobile apps are more prevalent than ever. This tremendous development has been made possible because smartphones have become much more popular and faster, with mobile networks developed to this point to allow us to always stay connected to the web.

Android and iOS are two traditional mobile operating systems. However, when developing an app for your business, there is always a big challenge regarding which platform to use to get the best customer engagement. Both – **Flutter and React Native** are considered the fastest growing cross-platform development languages for building native mobile apps for iOS and Android devices. These are two cross-platform mobile frameworks with great features, powerful user communities, and many pre-programmed modules that help you increase your development speed.

Without knowing the differences between React Native and Flutter, you can't make the right decision to complete your next project on the proper foundation. Whatever app you want to develop, whether it's a transportation app, medical app, on-demand delivery app, or music app, you need to know all of them. Your project requires carefully choosing a frame. It can be challenging to choose between Flutter and React Native. The most popular cross-platform development choice is React Native, developed by Facebook, but now there is a new entry on the table, namely Flutter, developed by Google. Admittedly, Flutter is not yet so mature, but its first tests imply that.

To resolve your confusion, we'll compare these two frameworks – **Flutter vs. React Native** so that it becomes clear which framework will serve you best for your next development project. It is essential to weigh these two options, understand their pros and cons, and choose an implementation path that meets your needs.

## Flutter vs. React Native – Overview

### Flutter

Released in 2018, Flutter is a cross-platform user interface system designed by the Google community to accelerate the growth and development of engaging native apps. Flutter is free to use; it uses an object-oriented programming language known as Dart. It is considered to be one of the leading open-source frameworks with a complete development ecosystem.

### Flutter – Use Case

- MVP mobile apps.
- Applications that work with operating system-level functionality and hardware design.
- Adaptable user interface with high-level widgets.
- Advanced OS plugins with simplistic logic.
- Reactivate apps with extensive data integration.

### Popular apps created with Flutter

- Google Ads
- Ali Baba
- eBay
- Tencent
- Bmw

### Some notable stats on Flutter:

- About 68.8% of developing countries love Flutter.
- 7.2% of the community choose Flutter because of the popularity of its libraries and tools.
- About 39% of people use Flutter to build cross-platform mobile apps.
- The Play Store alone now has over 150,000 Flutter apps.

## **React Native**

React Native is an open-source JavaScript framework created by Facebook for cross-platform mobile development – Android and iOS apps. React Native allows developers to use its power by coupling it with different features of each platform, seamlessly merging desired features. In addition, react Native will enable developers to write applications in Java, JavaScript, Swift, and Objective-C.

### **React Native – Use Case**

- Simple cross-platform applications.
- Applications with a simple user interface.
- Rapid prototype applications.
- Applications with reusable elements.
- Applications that work with synchronous APIs.

### **Popular apps built with React Native**

- Walmart
- Facebook
- Instagram
- Bloomberg
- SoundCloud

### **Some remarkable statistics on React Native:**

- Around 40% of developers in 2020 are said to have used React Native for developing cross-platform applications.
- About 58% of the developing community prefer to use React Native.
- About 11.5% of developers choose React Native for its essential libraries and tools.

# Flutter vs. React Native – Pros and Cons

## Benefits of Flutter

**Hot Reloading** the platform allows the developer to view the applied changes in real-time without losing the application's state; it saves time and increases productivity and efficiency.

### 1. Rich-widgets

The most notable feature of Flutter is the way it uses out-of-the-box applications, ensuring a consistent development and design model. In addition, it provides rich Google-based widgets; they produce high-quality code and perform better compared to other open-source frameworks.

### 2. Seamless integration

It integrates smoothly with Objective C or Swift for iOS and Java for Android. Furthermore, it indicates that there is no need to rewrite the code.

### 3. Code sharing

The codes that are written are easily shared between platforms, which makes it excellent for MVP development.

### 4. Customized design

Flutter includes a high-performance, open-source graphics engine called Skia, allowing users to produce custom-designed apps that deliver pretty engaging experiences on Android and iOS devices.

### 5. Fast shipping

This feature saves build time with fast iteration cycles, as testing is only needed for a single code base.

## Disadvantages of Flutter

### 1. No third-party libraries

Flutter is one of the latest platforms for mobile app development; it continues to grow and therefore offers limited choices for libraries and third-party tools.

### 2. Large file size

This is another downside of Flutter, having a large file size of its applications. On the other hand, having a small file size gives you improved runtime and performance. In addition, mobile users need sufficient memory space to store larger applications.

### 3. Skills required

Flutter requires programmers to learn DART first. With this requirement for an additional learning phase, Flutter can increase the cost and time of any project.

### 4. Updates

Unable to immediately push updates and fixes into applications bypassing standard release processes.

## Benefits of React Native

**Native rendering**React Native uses a host platform to render APIs without any CSS or HTML markup requirements natively.

### 1. Optimal performance

React Native interacts with native Android and iOS elements and goes a step further to produce native API codes free of interference. Using threads that are precisely different from native APIs and UIs in React Native improves performance.

## **2. Ecosystem**

Leverages a robust ecosystem and UI libraries to render the app's appearance with every state change automatically.

## **3. Debugging**

React Native provides accessibility to intelligent debugging tools and error reporting.

## **4. Hot Reloading**

Hot Reloading allows the developer to change the source code to view the codes, even if they are not recompiling the application or adding new codes directly into a live application.

## **5. Modular architecture and simplified user interface**

React native provides a modular design and architecture that allows developers to update apps immediately. In addition, it has a simple and less complex user interface and uses a software interface to eliminate unnecessary elements to achieve proper sequencing for building applications.

# **Disadvantages of React Native**

## **1. Complex user interface**

React Native may not work as the right choice for you if you need to produce a mobile app of various transitions, animations, and interactions. The native API rendering may not support some native UI components, making the UI feel a bit out of sync.

## **2. Tools and plugins**

Third-party libraries used for a more advantageous implementation may turn out to be obsolete.

## **3. Performance**

Lack of support for multiprocessing and parallel threading can slow performance.

#### 4. Compatibility and debugging issues

The Chrome debugger can be challenging to modify, inspecting codes and UI components with precision. It would help if you allowed its integration with Flipper to overcome this problem, as Flipper provides tools for debugging.

## Flutter vs. React Native – Detailed Comparison

In each case, when thinking about investing in mobile development, it is essential to consider the options and choose an implementation path that matches your needs. These influencing factors are evident to almost any business worldwide and would present a clear picture for making more informed decisions.

**Performance** Flutter and React Native are both open-source projects; these two frameworks promote hot Reloading. It increases production efficiency by reducing the need to pause and restart your apps to see updates. Flutter does not need a bridge to interact between native modules due to the availability of native elements by default but React Native does need a bridge, so React Native has poor performance compared to Flutter.

### Mobile performance

In React Native, source code execution must bridge the gap between JavaScript code and the device's native ecosystem, enabling communication and data transfer. On the other hand, Flutter compiles to the native x86 and ARM libraries without any additional layers. As a result, it allows it to use fewer resources and time to execute code. Additionally, Flutter has a C ++ graphics rendering engine, which means animations run smoother and faster in this cross-platform solution.

### Web performance

React Native is considered the boss when it comes to website development. JavaScript is the primary front-end language alongside HTML and C ++, producing excellent performance on the web. Therefore, React Native is the most appropriate option to choose if site performance is important to your business, as React Native offers technology designed explicitly for the web.

## **1. Design and graphics**

React Native acquires native visual components and appearance that provide a personalised and seamless experience. In addition, when updating the operating system, the application elements are updated individually to maintain the similarity and usability with other native applications.

Flutter simplifies the implementation of a conventional neutral style for all devices to save on project scope. In Flutter, apps look the same regardless of device model and operating system version. In addition, navigation and other elements remain unchanged unless they are intentionally upgraded.

## **2. Demand and cost**

React Native's framework is based on JavaScript; several developers already know it, making it easier to get the developer solution. On the other hand, [Flutter developers](#) charge relatively less; it's pretty difficult to get one from you. It is more prevalent among the current generation of developers than the more sophisticated ones.

## **3. Reliability**

However, both of these frameworks are made by the two most enormous tech titans globally with unauthorised licences. As a result, Flutter is relatively easy and resistant to changes produced by operating system updates. On the other hand, React Native relies heavily on native components for additional adaptation efforts in system updates.

## **4. Compatibility, application features**

Flutter supports several recommendations on Android and iOS devices. Many times you might need native development for React Native and Flutter for excellent functionality. Although the writing part can solve some difficulties, it ultimately leads to reduced maintenance.

## **5. Cartography and geolocation**

The Google team offers a variety of plugins for Flutter. These plugins provide a positive experience of the tracking features in the apps. While React Native works well with one-off

tracking, some issues can still arise with constant monitoring that can only be resolved by writing parts in the device's native programming language.

## 6. Security features

These two technologies offer high levels of security for mobile applications to meet regulatory criteria. However, it is recommended to use fewer third-party libraries. In addition, it is always better to store less data on the client-side. Finally, additional login verification is required when the user exits the program or locks the screen because these authentication tokens may expire.

## 7. Community support

A large community of contributors is essential for success in open source development. The point that React Native gives to these installations is exceptional. This framework quickly gained popularity among Facebook developers. You can easily get information on React Native's various social media. Flutter is smaller than React Native but grows much faster. It is also increasingly popular. To get the best of the Flutter vs. React Native community, use these Google Flutter apps!

## Final Verdict – Flutter vs. React Native: which one to choose?

After analyzing several parameters of these frameworks, **Flutter vs. React Native** still doesn't have a clear winner. Because every project is different, choosing the proper framework depends entirely on your project parameters and business requirements. Nevertheless, flutter and React Native frameworks are excellent frameworks that can help you develop a cross-platform application faster and easier than any other majority of native tools available today.

These two frameworks have distinct advantages and disadvantages. There is no one precise choice that works for everyone. It all depends on the specific requirements of your organization. In general terms, for larger applications that require native programming, Flutter is the most suitable option. On the other hand, react Native adapts well to applications that can develop from plug-and-play modules. React Native may be a more sensible choice if you already have a team of skilled JS developers. Additionally, choose Flutter as an option when you need a striking and highly personalized design or are already familiar with Dart rather than JavaScript.

## **Frequently Asked Questions:**

### **1) Why should you use Flutter or React Native?**

React Native and Flutter are two cross-platform application development solutions specifically designed to accommodate the requirements of today's businesses. Choosing either of the two would ensure:

- > Faster deployment
- > Lower costs
- > Reusable code
- > Multiple platform support

### **2) Is React Native or Flutter future-proof?**

These two frameworks are the product of two technological titans, namely Facebook and Google. In addition, regular updates are provided to keep both platforms adapting to changes and making them stand the test of time. Both of these would soon become long-term developer favourites.

### **3) Which one has better UI / UX – Flutter or React Native?**

React Native relies on third-party libraries to access most of its native modules, while the Flutter framework comprises UI rendering elements and APIs for device access, navigation, and testing. Therefore, you should always have a better user interface; choose Flutter.

### **4) Which one is more reliable and why – React Native or Flutter?**

React Native relies heavily on native components for additional adaptation efforts in system changes, making it less secure and less reliable. On the other hand, Flutter is a basic framework resistant to changes brought by updates to the operating system, so it offers more reliability and security.